Machine Learning:
Classification/Supervised Learning: Perceptron

Perceptron

Many classification methods are *naturally defined* for the case when there are only *two categories* in the set of category labels. Such situations are usually called binary classification.

One of the simplest *binary classifiers* is perceptron.

Definition. Let $X = \{\bar{x_1}, \ldots, \bar{x_n}\}$ be a set of data points, where each point $\bar{x_i} = (a_1, \ldots, a_d)$ is a vector of length d. Let $C = \{+1, -1\}$ is the set of category labels, and let $Y = \{y_1, \ldots, y_n\}$, $y_i \in C$ be the category labels $y_i = class(\bar{x_i})$.

A perceptron is binary linear classifier that consists of

1. a linear function
$$f(\bar{x}) = \sum_{j=1}^{d} w_j \cdot a_j$$
for some vector $\mathbf{w} = (w_1, \ldots, w_d)$ of *weights*,

2. a threshold value θ, and

3. a decision procedure:
$$class(\bar{x}) = \begin{cases} +1 & \text{if } f(\bar{x}) > \theta; \\ -1 & \text{if } f(\bar{x}) < \theta; \end{cases}$$

Intuition. The perceptron function $f(\bar{x}) = \mathbf{w} \cdot \bar{x}$ defines a $d-1$ dimensional hyperplane through the d-dimensional feature space. Points on the *positive* side of f are classified into the *positive class* (the $+1$ class). Points on the negative side of f are classified to the negative class (the -1 class).

Notes.

- For a perceptron to correctly classify the data, the data must be *linearly separable*. A dataset is called *linearly separable* if there exists a hyperplane through its feature space that separates the points in one category from the points in another category.

- If there are multiple hyperplanes that linearly separate the data, the perceptron will converge to *one of them*. The error function for the perceptron is essentially
$$Error(f(\bar{x})) = \sum_{i=1}^{n} |f(\bar{x_i}) - y_i|,$$
i.e. the number of incorrectly classified data points. Therefore $Error(f) = 0$ for **any** hyperplane f that linearly separates the dataset, and the perceptron does not differentiate between such hyperplanes.

Training Perceptron

We first present the perceptron training algorithm for $\theta = 0$.

1. Set $\mathbf{w} = (0, \ldots, 0)$.
2. Pick $\eta > 0$, the *learning rate* of the perceptron.
3. For each training example $(\bar{x}, y), \bar{x} \in X$ do:
   (a) $y' = \mathbf{w} \cdot \bar{x}$
   (b) if y' and y have the same sign, do nothing.
   (c) if y' and y have different signs:
   $$w := w + \eta \cdot y \cdot \bar{x}$$

To train the perceptron with an arbitrary value of θ:

- replace the vector $\mathbf{w} = (w_1, \ldots w_d)$ with the vector $\mathbf{w}' = (w_1, \ldots, w_d, \theta)$.
- replace every vector $\bar{x} \in X$, where $x = (a_1, \ldots, a_d)$ with the vector $\bar{x'} = (a_1, \ldots, a_d, -1)$.
- Train the perceptron using the algorithm above on the weights $\mathbf{w}'$ and feature vectors $X' = \{\bar{x'_1}, \ldots, \bar{x'_n}\}$.

Note: If you squint at it, the training process for the perceptron classifier should remind you of something. Hint: where else have you seen the *learning rate* parameter?

Indeed, this algorithm is a special case of *gradient descent/gradient ascent*.

When to stop

The training can stop if:

- All $\bar{x} \in X$ have been correctly classified (i.e., when classification error = 0).
- Failing that, perceptron training can be stopped in one of the following ways:
  - After *M* iterations for some number $M > n$.
  - After the following detection error:
  $$Error' = \frac{1}{2}\sum_{i=1}^{n} |\mathbf{w} \cdot \bar{x}_i \cdot (sign(\mathbf{w} \cdot \bar{x}_i) - y_i)|$$
  stops decreasing.
  (Note: $Error'$ computes the sum of distances from the separating hyperplane of all points that are misclassified. We need the $\frac{1}{2}$ normalizing factor because $|sign(\mathbf{w} \cdot \bar{x}_i) - y_i| = 2$ when the perceptron misclassifies a data point.)

References

[1] Jure Leskovec, Anand Rajaraman, Jeffrey D. Ullman, *Mining of Massive Datasets*, 2nd Edition, Cambridge University Press, 2014.

[2] Mohammed J. Zaki, Wagner Meira Jr., *Data Mining and Analysis: Fundamental Concepts and Algorithms*, Cambridge University Press, 2014.